In the Drawings:

A replacement sheet 1 including FIG. 1A including a PRIOR ART legend is submitted herewith for substitution for FIG. 1A as originally filed.

REMARKS

Claims 1-21 stand rejected under 35 USC §101, as being directed to non-statutory subject matter. Claims 1-21 stand rejected under 35 USC §102(b), as being anticipated by Rosenberg publication "Dbug".

The abstract of the disclosure has been amended to overcome the objections to the abstract. A replacement FIG. 1A of the drawings is submitted herewith having the label PRIOR ART added to overcome the objections to the drawings.

Claims 1, 3, 8, 9, 14, 17 and 18 have been amended to more clearly state the invention.

Reconsideration and withdrawal of the rejection of claims 1-21, as amended, under 35 USC §101, as being directed to non-statutory subject matter, is respectfully requested. As amended, each of the independent claims 1, 8, 14, and 17, now defines that the graphical debugger or debugger program product include instructions stored on a computer readable storage medium, said instructions when executed by the computer system to cause the computer system perform the step or implement functions. Thus, withdrawal of the rejection under 35 USC §101 of claims 1-21, as amended, is respectfully requested.

Reconsideration and allowance of each of the claims 1-21, as amended, is respectfully requested.

The present invention solves problems with the use of conventional debugger programs where some programs dynamically load program objects and

object archives at run time. The disclosed debuggers of Dbug, like other conventional debuggers, display a flat list of source files that are in the program being debugged. Although this is very useful, sometimes the programmer needs additional data that cannot be displayed with known debuggers. For programs that dynamically bind to program objects, the information about these objects is stored in a loadmap. Today the only other way to discern this information would be to examine make files, or to dump the debug data of the individual object files or archives with an external dumping tool. The present invention provides an improved debugger having a mechanism when debugging such programs enabling display for the programmer of what objects are in each part of the loadmap and the source files that correlate to these objects.

For a claim of a patent to be "anticipated" each and every element of that claim must be disclosed in a single prior art reference. Lack of novelty can be established only where a prior invention is identical to (or "anticipates") the invention sought to be patented. "In addition, the prior art reference must be enabling, thus placing the allegedly disclosed matter in the possession of the public." Akzo N.V. v. U.S. Intern. Trade Com'n, 808 F.2d 1471, 1479 (Fed. Cir. 1986). The elements found in the prior art reference must be arranged as in the claim. Connell v. Sears, Roebuck & Co., 722 F.2d 1542, 1548 (Fed. Cir. 1983).

Each of the independent claims 1, 8, 14, and 17, as amended, is patentable of the references of record including the Dbug publication. As amended, the steps of generating and the program loadmap is further defined to include a list of program objects being bound to the program under debug at runtime and the steps of

displaying the program loadmap includes displaying a program loadmap including said generated list of program objects being bound to the program under debug at runtime. These features of the invention are only taught by applicants, Dbug does not enable, nor provide any suggestion of such debugger features as taught and claimed by applicants, as recited in each of the independent claims 1, 8, 14, and 17, as amended.

Dbug does not teach or suggest a debugger function or step of generating a list of program objects being bound to the program under debug at runtime. The disclosed debugger for Java applications is not equivalent to and does not achieve the subject matter of the debugger features as taught and claimed by applicants, as recited in each of the independent claims 1, 8, 14, and 17, as amended. Dbug does not teach or suggest the steps of displaying the program loadmap that includes said generated list of program objects being bound to the program under debug at runtime, as recited in each of the independent claims 1, 8, 14, and 17, as amended.

Anticipation is established only when a single prior art reference discloses, expressly or under principles of inherency, each and every element of a claimed invention. Leinoff v. Louis Milona & Sons, Inc., 726 F.2d 734, 220 U.S.P.Q. 845 (Fed. Cir. 1984). Dbug does not disclose each and every element of claim 1, and Dbug does not enable the elements of claim 1 and fails to place the subject matter of the invention, as recited in claim 1 in the possession of the public. Each of the independent claims 8, 14, and 17, as amended, is patentable for the same reasons as claim 1.

Each of the independent claims 1, 8, 14, and 17, as amended, is patentable.

Each of the dependent claims 2-7, 9-13, 15-6 and 18-21, as amended, depends from respective patentable claims 1, 8, 14, and 17, as amended, further defining the invention. Thus, each of the dependent claims 2-7, 9-13, 15-6 and 18-21 is likewise patentable.

Applicants have reviewed all the art of record, and respectfully submit that the claimed invention is patentable over all the art of record, including the references not relied upon by the Examiner for the rejection of the pending claims.

It is believed that the present application is now in condition for allowance and allowance of each of the pending claims 1-21, as amended, is respectfully requested. Prompt and favorable reconsideration is respectfully requested.

If the Examiner upon considering this amendment should find that a telephone interview would be helpful in expediting allowance of the present application, the Examiner is respectfully urged to call the applicants' attorney at the number listed below.

Respectfully submitted,

By: Joan Pennington

Joan Pennington
Reg. No. 30,885
Telephone: (312) 670-0736